A DIVERSE LARGE-SCALE DATASET FOR EVALUATING REBROADCAST ATTACKS

Shruti Agarwal, Wei Fan, Hany Farid

Department of Computer Science, Dartmouth College, Hanover, NH 03755, USA
{shruti.agarwal.gr, wei.fan, hany.farid}@dartmouth.edu

ABSTRACT

We describe the acquisition of a large, diverse set of rebroadcast images captured by a screen-grab, scanning a printed photo, or rephotographing a displayed or a printed photo. This dataset consists of 14,500 rebroadcast images captured from a diverse set of devises: 234 displays, 173 scanners, 282 printers, and 180 recapture cameras. The diversity of this dataset—across devices and types of rebroadcast—poses significant challenges to detecting rebroadcast attacks. We evaluate the efficacy of four different classifiers trained to simultaneously detect all types of rebroadcast images.

Index Terms— Rebroadcast Attack, Recapture Attack, Digital Forensics, Biometrics

1. INTRODUCTION

A broad range of file-based forensic techniques have proven to be effective at detecting modifications of an original digital JPEG file [1]. These include analyzing JPEG compression parameters, JPEG file markers, and EXIF format and content [2, 3], analyzing sensor noise patterns [4] and sensor color filter array patterns [5, 6], and analyzing the underlying discrete cosine transform coefficients for evidence of multiple compressions [7–9]. Despite their efficacy, these techniques suffer from a simple rebroadcast attack in which an altered image is simply re-imaged, thus ensuring that any underlying camera properties are preserved. Rebroadcast content can also be used to attack biometric systems [10].

There are four standard types of rebroadcast attack generated by: (1) photographing a printed copy of an image; (2) scanning a printed copy of an image; (3) photographing a displayed image; or (4) capturing a screen-grab of a displayed image. Some of these rebroadcast images may require some further manipulation to add the necessary image metadata to be consistent with a camera original.

Many techniques have been developed to detect rebroadcast attacks. These include the use of higher-order wavelet statistics to identify scanned images [11], local binary patterns to identify displayed images [12], Markov-based features to identify printed images [13], physics-based features to identify printed images [14], noise statistics and JPEG compression to identify displayed images [15], aliasing patterns to identify displayed images [16], image-edge profiles to identify displayed images [17], and a convolutional neural network to identify displayed images [18]. Each of these studies detected only a single type of rebroadcast attack. In contrast, a few other techniques attempt to simultaneously de-

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA FA8750-16-C-0166). The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

1. INTRODUCTION

A broad range of file-based forensic techniques have proven to be effective at detecting modifications of an original digital JPEG file [1]. These include analyzing JPEG compression parameters, JPEG file markers, and EXIF format and content [2, 3], analyzing sensor noise patterns [4] and sensor color filter array patterns [5, 6], and analyzing the underlying discrete cosine transform coefficients for evidence of multiple compressions [7–9]. Despite their efficacy, these techniques suffer from a simple rebroadcast attack in which an altered image is simply re-imaged, thus ensuring that any underlying camera properties are preserved. Rebroadcast content can also be used to attack biometric systems [10].

There are four standard types of rebroadcast attack generated by: (1) photographing a printed copy of an image; (2) scanning a printed copy of an image; (3) photographing a displayed image; or (4) capturing a screen-grab of a displayed image. Some of these rebroadcast images may require some further manipulation to add the necessary image metadata to be consistent with a camera original.

Many techniques have been developed to detect rebroadcast attacks. These include the use of higher-order wavelet statistics to identify scanned images [11], local binary patterns to identify displayed images [12], Markov-based features to identify printed images [13], physics-based features to identify printed images [14], noise statistics and JPEG compression to identify displayed images [15], aliasing patterns to identify displayed images [16], image-edge profiles to identify displayed images [17], and a convolutional neural network to identify displayed images [18]. Each of these studies detected only a single type of rebroadcast attack. In contrast, a few other techniques attempt to simultaneously de-

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA FA8750-16-C-0166). The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

1. INTRODUCTION

A broad range of file-based forensic techniques have proven to be effective at detecting modifications of an original digital JPEG file [1]. These include analyzing JPEG compression parameters, JPEG file markers, and EXIF format and content [2, 3], analyzing sensor noise patterns [4] and sensor color filter array patterns [5, 6], and analyzing the underlying discrete cosine transform coefficients for evidence of multiple compressions [7–9]. Despite their efficacy, these techniques suffer from a simple rebroadcast attack in which an altered image is simply re-imaged, thus ensuring that any underlying camera properties are preserved. Rebroadcast content can also be used to attack biometric systems [10].

There are four standard types of rebroadcast attack generated by: (1) photographing a printed copy of an image; (2) scanning a printed copy of an image; (3) photographing a displayed image; or (4) capturing a screen-grab of a displayed image. Some of these rebroadcast images may require some further manipulation to add the necessary image metadata to be consistent with a camera original.

Many techniques have been developed to detect rebroadcast attacks. These include the use of higher-order wavelet statistics to identify scanned images [11], local binary patterns to identify displayed images [12], Markov-based features to identify printed images [13], physics-based features to identify printed images [14], noise statistics and JPEG compression to identify displayed images [15], aliasing patterns to identify displayed images [16], image-edge profiles to identify displayed images [17], and a convolutional neural network to identify displayed images [18]. Each of these studies detected only a single type of rebroadcast attack. In contrast, a few other techniques attempt to simultaneously de-

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA FA8750-16-C-0166). The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

1. INTRODUCTION

A broad range of file-based forensic techniques have proven to be effective at detecting modifications of an original digital JPEG file [1]. These include analyzing JPEG compression parameters, JPEG file markers, and EXIF format and content [2, 3], analyzing sensor noise patterns [4] and sensor color filter array patterns [5, 6], and analyzing the underlying discrete cosine transform coefficients for evidence of multiple compressions [7–9]. Despite their efficacy, these techniques suffer from a simple rebroadcast attack in which an altered image is simply re-imaged, thus ensuring that any underlying camera properties are preserved. Rebroadcast content can also be used to attack biometric systems [10].

There are four standard types of rebroadcast attack generated by: (1) photographing a printed copy of an image; (2) scanning a printed copy of an image; (3) photographing a displayed image; or (4) capturing a screen-grab of a displayed image. Some of these rebroadcast images may require some further manipulation to add the necessary image metadata to be consistent with a camera original.

Many techniques have been developed to detect rebroadcast attacks. These include the use of higher-order wavelet statistics to identify scanned images [11], local binary patterns to identify displayed images [12], Markov-based features to identify printed images [13], physics-based features to identify printed images [14], noise statistics and JPEG compression to identify displayed images [15], aliasing patterns to identify displayed images [16], image-edge profiles to identify displayed images [17], and a convolutional neural network to identify displayed images [18]. Each of these studies detected only a single type of rebroadcast attack. In contrast, a few other techniques attempt to simultaneously de-

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA FA8750-16-C-0166). The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

1. INTRODUCTION

A broad range of file-based forensic techniques have proven to be effective at detecting modifications of an original digital JPEG file [1]. These include analyzing JPEG compression parameters, JPEG file markers, and EXIF format and content [2, 3], analyzing sensor noise patterns [4] and sensor color filter array patterns [5, 6], and analyzing the underlying discrete cosine transform coefficients for evidence of multiple compressions [7–9]. Despite their efficacy, these techniques suffer from a simple rebroadcast attack in which an altered image is simply re-imaged, thus ensuring that any underlying camera properties are preserved. Rebroadcast content can also be used to attack biometric systems [10].

There are four standard types of rebroadcast attack generated by: (1) photographing a printed copy of an image; (2) scanning a printed copy of an image; (3) photographing a displayed image; or (4) capturing a screen-grab of a displayed image. Some of these rebroadcast images may require some further manipulation to add the necessary image metadata to be consistent with a camera original.

Many techniques have been developed to detect rebroadcast attacks. These include the use of higher-order wavelet statistics to identify scanned images [11], local binary patterns to identify displayed images [12], Markov-based features to identify printed images [13], physics-based features to identify printed images [14], noise statistics and JPEG compression to identify displayed images [15], aliasing patterns to identify displayed images [16], image-edge profiles to identify displayed images [17], and a convolutional neural network to identify displayed images [18]. Each of these studies detected only a single type of rebroadcast attack. In contrast, a few other techniques attempt to simultaneously de-

This research was developed with funding from the Defense Advanced Research Projects Agency (DARPA FA8750-16-C-0166). The views, opinions, and findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.
Table 1. Total number of (b) rebroadcast images along with a breakdown in the diversity of the (c) original recording device, the (d-e) rebroadcast medium, and the (f-g) recapture device.

<table>
<thead>
<tr>
<th>(a) Rebroadcast Type</th>
<th>(b) Image Count</th>
<th>(c) Original Camera Count</th>
<th>(d) Rebroadcast Medium</th>
<th>(e) Rebroadcast Medium Count</th>
<th>(f) Recapture Device</th>
<th>(g) Recapture Device Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>print</td>
<td>2,824</td>
<td>998</td>
<td>printer</td>
<td>148</td>
<td>digital camera</td>
<td>109</td>
</tr>
<tr>
<td>scan</td>
<td>3,821</td>
<td>990</td>
<td>printer</td>
<td>182</td>
<td>flatbed scanner</td>
<td>173</td>
</tr>
<tr>
<td>display</td>
<td>3,873</td>
<td>1,036</td>
<td>display</td>
<td>129</td>
<td>digital camera</td>
<td>119</td>
</tr>
<tr>
<td>screen-grab</td>
<td>3,975</td>
<td>1,021</td>
<td>display</td>
<td>132</td>
<td>NA</td>
<td>NA</td>
</tr>
</tbody>
</table>

Workers were also asked to report the make and model of all devices used in completing their task. To ensure high quality submissions, the workers were required to photograph the printed/displayed images with minimum perspective distortion and to use the maximum imaging resolution afforded by the recapturing device. For the print, scan, and display images, the rebroadcast images were saved in the JPEG format. The screen-grab images were saved in the PNG format. AMT workers were asked to submit the rebroadcast images without any further modifications.

Workers were given a variety of instructions to help us ensure the validity of the submitted images. For print and display images, workers were asked to frame the image so that the background was clearly visible at the image boundary. For the screen-grab images, workers were similarly asked to include part of their desktop background. For the scan images, workers signed their name on the boundary of the printed image. Each submitted image was manually reviewed to make sure that they satisfied all of the required criteria, was manually cropped to remove the extraneous boundary, and saved as a PNG image (to avoid double-compression artifacts).

Starting with 10,000 original images, we collected a total of 14,500 rebroadcast images. An additional 4,500 original images (without a rebroadcast version) were added to the final dataset. Shown in Table 1 is the breakdown of total images for each rebroadcast type as well as the number of unique imaging devices used in each category. Across all rebroadcast types, the collected images span 1,294 original cameras, 234 different displays (for display and screen-grab), 173 different scanners, 282 different printers (for print and scan), and 180 different recapture cameras (for print and display).

For each rebroadcast type, images were acquired using a wide range of imaging configurations – defined as a unique original camera, rebroadcast medium, and recapture device combination. There were a total of 2,658 imaging configurations used for print images, 3,666 for scan images, 3,735 for display images, and 3,737 for screen-grab images.

3. METHODS

We describe three standard feature sets in combination with an SVM, and a CNN-based approach to distinguish original from rebroadcast images.

3.1. Local Binary Pattern Based Feature

Local binary pattern (LBP) has been used to represent local image texture for image analysis [24]. An LBP-based texture feature \((L_{P,R})\), for a monochrome image, consists of a normalized occurrence histogram of texture patterns in a local neighborhood. The values \(P\) and \(R\) determine the dimensionality and scale of the features: \(P\) is the number of neighboring pixels selected at a radius \(R\) from the feature center. Variants of LBP texture features have been used to identify rebroadcast attacks [12, 19, 20].

For our tests, we implemented the approach described in [12]. This approach yields a feature dimensionality of 80. As described in Section 4, a non-linear SVM was employed to simultaneously distinguish original from all four types of rebroadcast images.

3.2. Multi-Scale Wavelet Statistic Based Feature

Wavelet decomposition [25] of images has found wide-spread applications in the domain of image representation. The wavelet decomposition represents an image in terms of oriented spatial frequency subbands. For natural images, the distribution of wavelet coefficients in each subband is well modeled with a generalized Laplacian [26]. With the assumption that distortions to a natural image will disrupt these natural image statistics, unnatural manipulations like a rebroadcast attack can be identified as proposed in [11, 12].

For our tests, we implemented the approach described in [12]. This approach yields a feature dimensionality of 54. A non-linear SVM was employed to simultaneously distinguish original from all four types of rebroadcast images.

3.3. Markov-Based Feature

Markov chains have been used often in steganalysis to capture the statistics of natural images in both spatial and frequency domains [27–29]. As described in [13, 28], Markov-based features are computed for a monochrome image by first applying a 2-D discrete cosine transform (DCT) to every non-overlapping \(8 \times 8\) block. The resulting DCT coefficients are then converted from floating-point to integer values. Four difference arrays are generated by computing the difference of each DCT coefficient with its neighboring coefficient in the horizontal, vertical, and two diagonal directions. Each array is then modeled as a Markov random process using a one-step transition probability matrix [30].

For our tests, we implemented the approach described in [13]. This approach yields a feature dimensionality of 196. As before, a non-linear SVM was employed to simultaneously distinguish original from all four types of rebroadcast images.

3.4. Convolutional Neural Network

Training a convolutional neural network (CNN) on full-resolution images imposes significant demands on computational costs and data acquisition. We, therefore, train our network on 64 \(\times\) 64 image blocks. A monochrome image is partitioned into non-overlapping
Fig. 1. Shown is our proposed CNN architecture. Each rectangular block corresponds to a feature map: the number of channels and the dimension of the feature maps is denoted below and along the sides of each block, respectively. Between each feature map are multiple network layers: K3S \{1, 2\} denotes a convolutional layer with kernel size 3 × 3 and stride 1 or 2, LReLU denotes a leaky ReLU activation, BN denotes batch normalization, and FC denotes a fully connected layer. The input to the network is a monochrome 64 × 64 image block that is pre-processed with 16 pre-defined filters. The network outputs a two-dimensional vector that is used to classify an image as original or rebroadcast.

64 × 64 blocks from which a maximum of 300 blocks with the highest spatial variance are extracted. This selection is done because we find empirically that these high spatial frequency regions afford better classification accuracy.

Shown in Fig. 1 is our network architecture. The input (after pre-processing) to the network is a 16 × 64 × 64 tensor corresponding to the concatenation of 16 Gaussian filter residuals applied to a 64 × 64 image block. Eight of the filters are of size 3 × 3 and eight are of size 5 × 5, each with a standard deviation equally spaced in the log domain: 0.50, 0.58, 0.68, 0.80, 0.94, 1.10, 1.28, and 1.50.

Our network consists of six convolutional layers and two fully connected layers followed by one log-softmax layer. The output of each convolutional layer and the first fully connected layer is followed by a leaky rectified linear unit activation (ReLU). Each convolutional layer uses a 3 × 3 filter size and the number of filters increases with network depth as shown in Fig. 1. To stabilize the training, batch normalization is used after each leaky ReLU layer (except for the first and last layer).

The network outputs a two-dimensional vector \( \vec{v}^T = (v_1 \ v_2) \). The image input is classified as original if \( v_1 > v_2 \) and rebroadcast otherwise.

During training, the batch size is 64, the momentum is 0.9, and a cross entropy loss function is used. The learning rate is initialized to 0.001 and is decreased by a factor of 0.9 when the loss plateaus. The network was trained for two epochs (135,000 iterations in total). After every 1,000 iterations, the accuracy on the validation dataset is recorded. The final model is the model with the highest validation accuracy. Our network is implemented using the PyTorch framework [31].

Because our network is trained on 64 × 64 image blocks and not an entire image, we employ a simple voting scheme to classify an entire image as either original or rebroadcast. As in the training and validation, a maximum of 300 non-overlapping image blocks with the highest spatial variance are extracted from a full-size image. Each image block is classified as original or rebroadcast by our network. If more than \( T\% \) (50 ≤ T < 100) of the image blocks are classified as rebroadcast then the image is classified as rebroadcast. If more than \( T\% \) of the image blocks are classified as original then the image is classified as original. If neither of these cases is satisfied, then the image is not classified.

4. RESULTS

Using the three feature sets described in Sections 3.1-3.3 we train three separate SVMs [32] to simultaneously identify all types of rebroadcast images (print, scan, display, and screen-grab). The dataset of 14,500 original and 14,500 rebroadcast images is randomly divided into 80:20 training and testing datasets. A non-linear SVM with a radial basis kernel function (RBF) is trained using 5-fold cross validation to select the best values for the cost of mis-classification (c) and the RBF parameter (\( \gamma \)).

Shown in Table 2 are the true positive rates (i.e., correctly classifying a rebroadcast image) for these SVMs for a 0.1% and 1.0% false positive rate (mis-classifying an original image as rebroadcast). These accuracies correspond to the average accuracy over 100 random training/testing splits. For each feature set (LBP, wavelet, LBP+wavelet, Markov) we report the classification accuracy for our dataset (mturk+, with 29,000 images), our dataset combined with the four datasets described in Section 1 (mturk+, with 46,853 images), and separately for the four individual datasets.

These results illustrate the fragility of some techniques when trained against relatively small and homogeneous datasets. For example, the LBP features yield a 93.4% detection accuracy (with 0.1% false positive) when trained against the 2,700 image dataset of [12], but only a 4.9% detection accuracy when tested against our larger and more diverse dataset of 29,000 images. Similarly, the accuracy for the wavelet features drops from 74.2% to 4.5% on these same two datasets. As shown in the bottom few rows of Table 2, although the Markov features yield an accuracy of 88.9% on dataset [12] as compared to LBP’s 93.4%, the Markov features generalize much better yielding an accuracy of 82.0% on our dataset as compared to 4.9% for LBP. Overall, the Markov features significantly outperform the LBP, wavelet, or combined LBP and wavelet features.
Table 2. Image classification for SVM on six different datasets (the bracketed values correspond to the datasets specified in the given reference). Columns (a) and (b) correspond to the true positive rate for a false positive of 0.1% and 1.0%.

<table>
<thead>
<tr>
<th>Features</th>
<th>Dataset</th>
<th>True Positive (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>(a)</td>
</tr>
<tr>
<td>LBP</td>
<td>mturk</td>
<td>4.9</td>
</tr>
<tr>
<td></td>
<td>mturk+</td>
<td>4.1</td>
</tr>
<tr>
<td></td>
<td>[12]</td>
<td>93.4</td>
</tr>
<tr>
<td></td>
<td>[13]</td>
<td>45.7</td>
</tr>
<tr>
<td></td>
<td>[17]</td>
<td>53.6</td>
</tr>
<tr>
<td></td>
<td>[22]</td>
<td>25.4</td>
</tr>
<tr>
<td>wavelet</td>
<td>mturk</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td>mturk+</td>
<td>4.2</td>
</tr>
<tr>
<td></td>
<td>[12]</td>
<td>74.2</td>
</tr>
<tr>
<td></td>
<td>[13]</td>
<td>50.1</td>
</tr>
<tr>
<td></td>
<td>[17]</td>
<td>83.6</td>
</tr>
<tr>
<td></td>
<td>[22]</td>
<td>13.2</td>
</tr>
<tr>
<td>LBP+wavelet</td>
<td>mturk+</td>
<td>6.4</td>
</tr>
<tr>
<td></td>
<td>[12]</td>
<td>5.2</td>
</tr>
<tr>
<td></td>
<td>[17]</td>
<td>82.0</td>
</tr>
<tr>
<td></td>
<td>[22]</td>
<td>98.7</td>
</tr>
<tr>
<td></td>
<td>[22]</td>
<td>64.9</td>
</tr>
<tr>
<td>Markov</td>
<td>mturk</td>
<td>82.0</td>
</tr>
<tr>
<td></td>
<td>mturk+</td>
<td>64.0</td>
</tr>
<tr>
<td></td>
<td>[12]</td>
<td>88.9</td>
</tr>
<tr>
<td></td>
<td>[13]</td>
<td>89.6</td>
</tr>
<tr>
<td></td>
<td>[17]</td>
<td>99.6</td>
</tr>
<tr>
<td></td>
<td>[22]</td>
<td>94.9</td>
</tr>
</tbody>
</table>

Shown in Table 3 are the detection accuracies broken down by rebroadcast type for the mturk dataset and Markov features where it can be seen that there is no large difference in the detection accuracy across rebroadcast type.

Our CNN is trained, validated and tested on the 29,000 original and rebroadcast images described in Section 2, partitioned into 17, 400 training images, 5, 800 validation images, and 5, 800 testing images. A total of 4.35, 1.44, and 1.45 million blocks are extracted from the training, validation, and testing images, respectively.

Shown in Table 4 are the CNN testing accuracies. The first row corresponds to the mturk dataset and the second row corresponds to the mturk+ dataset, as described above. The classification threshold $T$ was set to yield a false positive rate of 0.1% ($T = 55\%$ for mturk and $T = 73\%$ for mturk+).

With a false positive rate of 0.1%, our network achieves a detection accuracy of more than 97% on both datasets. Note, however, that unlike the SVM results in Table 2, this network can occasionally fail to classify a small number of images (see last column of Table 4). This failure to classify is the result of the voting scheme used to translate detection of image blocks to detection of an entire image.

The results in Table 4 correspond to a detection accuracy on entire images. At the image block level, our network is tested on 1.45 million image blocks (mturk dataset) yielding a true positive rate of 98.8% and false positive rate of 0.7%. Similarly, for the mturk+ dataset with 2.4 million image blocks, the true positive rate is 97.7% with a false positive rate of 1.8%. Overall, the CNN significantly outperforms the more classic hand-crafted feature selection (see Table 2).

5. DISCUSSION

We have collected a diverse, large-scale dataset of images for the evaluation of rebroadcast attacks on forensic and biometric techniques. We are making this dataset available upon request.

Using this dataset, we have shown that some previous techniques for detecting rebroadcast attacks trained on smaller and more homogeneous datasets do not generalize to larger more diverse datasets. We hypothesize that this failure is because each step of a rebroadcast attack (the original imaging device, the rebroadcast medium and device, and the recapture device) introduces distinct image artifacts that are not properly captured in small homogeneous datasets.

We have also shown that both classic handcrafted features and neural networks are capable of simultaneously detecting multiple types of rebroadcast attacks. The handcrafted Markov-based features significantly outperform the other popular local binary pattern and wavelet features, but a neural network significantly outperforms all of these approaches. Although our network architecture yields good detection accuracy, we expect that modifications to this architecture may lead to further improvements.

We were somewhat surprised that a single classifier was able to simultaneously detect all four types of rebroadcast attack and are currently investigating the nature of the feature differences between original and rebroadcast that afford this type of generalization.

6. REFERENCES


Zhiwei Zhang, Junjie Yan, Sifei Liu, Zhen Lei, Dong Yi, and P. Yang, R. Ni, and Y. Zhao, “Recapture image forensics based on physics-based features,” in *International Conference on Multimedia and Ubiquitous Engineering*, vol. 8, no. 5, pp. 71–82, 2013.


Amazon mechanical turk,” https://www.mturk.com/.


